```
static const double    kOneThird       = (1.0 / 3.0);
static const double    kLabExp                 = 2.8;
static const double    kChromaPower    = 1.5;
static const double    kWhiteRefCoeff = 2.5;
static const double    kLabThresh              = (0.00885600);
static const double    kLabCbrtThresh = pow(kLabThresh,
       kOneThird);
static const double kBlackPtScale = -0.0;
#define LabGammaThresh(gamma) pow(kLabThresh, gamma)
#define min2_(x,y)              (x>y ? y : x)
#define max2_(x,y)              (x>y ? x : y)
#define min3_(x,y,z)    (min2_(x,y)> z? z : min2_(x,y))
#define max3_(x,y,z)    max2_(0.0001,(max2_(x,y)> z?
       max2_(x,y) : z))

void CalcLABPLUS( double X,double Y, double Z,
                  double Xm,double Ym, double Zm,
                  double Xn,double Yn, double Zn,
                  double *L_star, double *a_star, double
                  *b_star)

{
double X,Y,Z,Xm,Ym,Zm,Xn,Yn,YnL,Zn,Xr,Yr,Zr,cbrtY;
double ratX, ratY, ratZ;
double POWER_XYZ = 1.0/kLabExp;
static const double LAB_THRESH =((double) 0.00885600);
static const double LAB_CONST =((double) 0.13793103);
double Lscale = (116.0*pow(LAB_THRESH,POWER_XYZ) -
       16.0)/LAB_THRESH;
double XYZscale = (pow(LAB_THRESH,POWER_XYZ) -
        LAB_CONST)/LAB_THRESH;

double  deltaWhitePtX = Xm - Xn;
double  deltaWhitePtY = Ym - Yn;
double  deltaWhitePtZ = Zm - Zn;

double Xnorm = X/Xm;
double Ynorm = Y/Ym;
double Znorm = Z/Zm;

double XYZlength =
       sqrt(Xnorm*Xnorm+Ynorm*Ynorm+Znorm*Znorm);

double XYZDiagProjection = (Xnorm+Ynorm+Znorm)/sqrt(3.0);
```

```
double devXYZ = sqrt(XYZlength*XYZlength-
       XYZDiagProjection*XYZDiagProjection);
double maxDev = 2.0 * sqrt(3.0)/3.0;

devTangentXYZ = devXYZ/XYZDiagProjection;

whiteRefCorr = kWhiteRefCoeff *
       pow(devTangentXYZ/maxDev,kChromaPower);

double XAdj -= X - Xnorm * deltaWhitePtX * (1.0 - devXYZ);
double YAdj -= Y - Ynorm * deltaWhitePtY * (1.0 - devXYZ);
double ZAdj -= Z - Znorm * deltaWhitePtZ * (1.0 - devXYZ);

doubleXbm = Xm * kBlackPtScale;
doubleYbm = Ym * kBlackPtScale;
doubleZbm = Zm * kBlackPtScale;

Xn = (X/ XAdj) * Xn;
Yn = (Y/ YAdj) * Yn;
Zn = (Z/ ZAdj) * Zn;

Xr = X/Xn;
Yr = Y/ Yn;
Zr = Z/Zn;
cbrtY = pow(Yr, POWER_XYZ);

116.0*cbrtY - 16.0 : 903.3*Yr);
cbrtY = pow(Yr, POWER_XYZ);

cbrtY = (Yr > LAB_THRESH) ? cbrtY : XYZscale*Yr + LAB_CONST;

*a_star = 500.0 *(((Xr > LAB_THRESH) ? pow(Xr, POWER_XYZ) :
       XYZscale*Xr + LAB_CONST) - cbrtY);

*b_star = 200.0 *(cbrtY - ((Zr > LAB_THRESH) ? pow(Zr,
       POWER_XYZ) : XYZscale*Zr + LAB_CONST));

Yr = Y/ Ym;
cbrtY = pow(Yr, POWER_XYZ);
*L_Star =((Yr>LAB_THRESH)?116.0*cbrtY-16.0: Lscale*Yr);

return;
}
```